Price et al. *EPJ Quantum Technology* (2020) 7:8
https://doi.org/10.1140/epjqt/s40507-020-00084-6

EPJ Quantum Technology
a SpringerOpen Journal

EPJ.org

RESEARCH Open Access

A quantum key distribution protocol for rapid denial of service detection

Alasdair B. Price[1,2*], John G. Rarity[1] and Chris Erven[1]

*Correspondence:
alasdair.price@bristol.ac.uk
[1]Quantum Engineering Technology Labs, H. H. Wills Physics Laboratory & Department of Electrical and Electronic Engineering, University of Bristol, Bristol, UK
[2]Quantum Engineering Centre for Doctoral Training, H. H. Wills Physics Laboratory & Department of Electrical and Electronic Engineering, University of Bristol, Bristol, UK

Abstract

We introduce a quantum key distribution protocol designed to expose fake users that connect to Alice or Bob for the purpose of monopolising the link and denying service. It inherently resists attempts to exhaust Alice and Bob's initial shared secret and is 100% efficient, regardless of the number of qubits exchanged above the finite key limit. Additionally, secure key can be generated from two-photon pulses without having to make any extra modifications. This is made possible by relaxing the security of BB84 to that of the quantum-safe block cipher used for day-to-day encryption, meaning the overall security remains unaffected for useful real-world cryptosystems such as AES-GCM being keyed with quantum devices.

Keywords: Quantum Cryptography; Quantum Key Distribution; Denial of Service; Sifting; Photon Number Splitting

1 Introduction

Quantum key distribution (QKD) enables two remote parties (Alice and Bob) to generate a shared secret, using quantum mechanics to ensure security against all eavesdropping on an idealised quantum channel [1–3]. The resulting shared secret is guaranteed quantum-safe, making BB84 (the first QKD protocol, of which there are now a number of different variants) a strong candidate for niche applications where very high levels of security are required. Further advantage can be gained in the form of eavesdropper detection, by exploiting the disturbances introduced when an attacker measures the quantum states. Unfortunately, these disturbances can also be due to noise, though for security purposes they must always be attributed to eavesdropping, opening up the potential for a denial of service (DoS) attack that can be carried out simply by increasing the error rate on the quantum transmission line. While sometimes used as an argument against QKD [4], the risk of this happening is often overstated, as it requires an attacker to have physical access to the optical fibre between Alice and Bob and is easily detectable, so the development of large-scale networks will mitigate any damage by enabling the quantum signal to be redirected. It should be noted that increasing the error rate to an intolerable level is functionally equivalent to cutting the QKD link and so, in this scenario, the same solution will apply. However, there is another way of performing DoS, that does not require an adver-

Springer

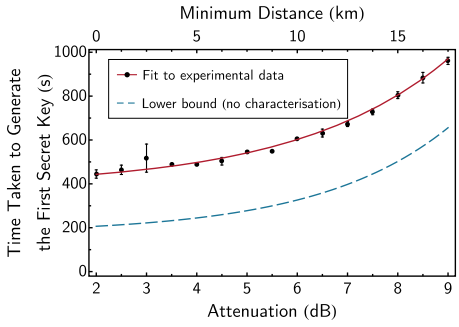

Figure 1 Time taken for a networked *ID Quantique Clavis*^{2} to generate a $\sim 10^5$-bit shared secret across a newly established connection. Each QKD system is connected to an optical switch (introducing 1 dB of loss in each case, hence a total attenuation of 2 dB at 0 km), allowing different links to be selected. The lower bound (calculated from average secret key rates) skips device and fibre characterisation, and assumes this does not affect the performance of subsequent steps, to give the shortest possible denial of service attack duration

sary to monitor all connections simultaneously and to which all current QKD protocols are vulnerable.

To prevent man-in-the-middle attacks, it is required that the classical QKD channel be authenticated and, to retain information-theoretic security, this must be done using a Wegman–Carter message authentication code (MAC) [5] keyed with a pre-shared secret. The MAC has to be transmitted at the end of the QKD protocol, authenticating every message sent up to that point [6], as authenticating each message individually would prohibit net positive key generation. This means neither Alice or Bob will know whether the person they are communicating with is genuine until they have a secret key, so an imposter could deny service to other users simply by opening a connection and performing QKD. Figure 1 shows how long this could last for, assuming only one round of key generation is carried out by the attacker. For a 10 km metropolitan-area network, the *ID Quantique Clavis*^{2} will communicate with an illegitimate party for roughly 10 minutes before realising! We note that the *Clavis*^{2} continues to work at attenuations above 9 dB but key generation starts to become intermittent. The average time taken for a successful round of QKD at 10 dB is close to 20 minutes, however the DoS impact could be greater if other rounds fail, which happens in over 30% of cases. Ultimately, it makes sense for an attacker to maximise the attenuation on their link to keep the systems occupied for as long as possible.

In this paper, we discuss how QKD can be modified to eliminate the risk of DoS attacks that leverage provably fake users. By this, we mean attacks where Eve initiates a QKD session between herself and either Alice or Bob, rather than attacks that require Eve to intervene on a QKD session between Alice and Bob. In the real world, cryptosystems that use QKD are unlikely to employ the one-time pad in day-to-day communications. Instead, quantum-safe ciphers such as the Advanced Encryption Standard (AES) [7] feature heavily [6, 8], as they utilise the key more efficiently. This means the overall system is not information-theoretically secure, so reducing the mathematical security of QKD in line with the encryption algorithm will not reduce the real-world security, and will actually increase it if DoS and side-channel attacks can be mitigated as a result. By making a few additional tweaks, we show that a computationally-secure QKD protocol can securely generate key even from singly detected two-photon terms, and run at exactly 100% efficiency.

2 Preliminaries

Authentication in QKD is traditionally performed using a Wegman–Carter MAC [5, 6, 8]. This takes the form

$$\tau = h_{k_{\mathrm{H}}}(m) \oplus k_{\mathrm{M}}, \tag{1}$$

where h is a universal hash function keyed with k_{H} (a bit string that forms one part of the initial shared secret), m is the message to be authenticated (in this case, a concatenation of every transmission made over the public channel), $\oplus$ is the XOR operation and k_{M} (a bit string that forms the other part of the initial shared secret) is the key used to mask the output of the hash. Alice calculates the tag τ for the information she publicly announced and sends it to Bob. He then computes the tag for the information he received and compares it with Alice's tag. So long as the two are the same, he can be confident that the information has not come from or been modified by a third party (Eve). The same can then be done for the messages sent from Bob to Alice.

As described in the Introduction, this way of handling QKD authentication creates the opportunity for an attacker to carry out a DoS attack, which we now formalise. It should be emphasised that, although the following requires Eve to have access to Alice and Bob's network, she does not need to be physically co-located with a specific optical fibre, as would be the case for the traditional DoS attack, also in the Introduction. Eve begins by establishing a high-loss connection with Alice and performing low bit rate QKD up to the point where she fails the authentication. While the session is active, Alice cannot exchange key with legitimate users such as Bob, meaning this constitutes denial of service of the Alice QKD unit (referred to as attack 1 hereafter). Eve's incursion can be prolonged if agents of Eve are queued behind her, turning it into a distributed denial of service (DDoS) attack. During this period, Alice and Bob are unable to generate new shared keys, which may also lead to denial of service of their classical communications. In addition, after succumbing to the above, Alice and Bob may find that they have exhausted their supply of pre-shared secret. This, a well-established vulnerability that also has the potential to be exploited independently, is known as a key exhaustion attack, which can be counteracted by using a post-quantum public-key algorithm to authenticate the next round of QKD [9, 10]. So long as Eve cannot break said algorithm in the short amount of time for which it is useful to her, full security is retained for all keys thereafter. However, by taking this approach, we have introduced a primitive that was not already part of the system, assuming Alice and Bob's initial secret was shared without using post-quantum cryptography. The recovery mechanism can also be triggered relatively easily, allowing attack 1 to be used as a way of forcing public-key algorithms to be used for every successful round of QKD. Therefore, from both simplicity and security perspectives, a reactive strategy is less than ideal.

Finally, for completeness, we should consider what would happen if a man in the middle were able to compromise the chosen authentication scheme. As Wegman–Carter MACs are unconditionally secure, a break of this nature is not considered possible for canonical BB84, assuming Eve does not have access to the initial secret key. However, it will be relevant later on when discussing the use of AES in both the QKD authentication and data encryption. Here, Eve intercepts the quantum bits (qubits), measures each one in a random basis and resends the results she observed in the bases she measured. She conceals her involvement by modifying Alice's bases announcement and Bob's response, along with the

authentication tags for each (we call this attack 2). Eve can now read all communications encrypted and/or authenticated using the key she shares with Alice, before forwarding them with or without modification, having re-encrypted or authenticated using the key she shares with Bob.

3 The protocol

We begin by trying to fulfil the main objective of this paper; preventing attack 1. A trivial solution, which preserves the information-theoretic security of BB84, would be to implement some form of access control that requests Eve verify her identity before she is allowed to connect. However, if there are no further checks until the end of the protocol, this could easily be circumvented by Eve switching out Bob for herself once key generation begins. Therefore, the most sensible approach is to authenticate every message exchanged by Alice and Bob.

Ideally, this will mean modifying equation (1) such that the tags can be reused without increasing the risk of an attacker being able to decrypt messages that rely on quantum keys. Brassard proposed in [11] that k_{M} could be defined as the output of a random function. In practice, this can be the cipher used for the data encryption, independently keyed with k_{C}, so we rewrite equation (1) as

$$\tau_i = h_{k_{\mathrm{H}}}(m_i) \oplus \mathrm{AES}_{k_{\mathrm{C}}}(s_i), \tag{2}$$

where s_i is a public one-time number, or "nonce". This time, the initial shared secret is comprised of k_{H} and k_{C}. A number of efficient authentication schemes such as poly1305-AES [12], UMAC [13] and VMAC [14] take this form (though their moduli for addition vary), and their security when accompanying a known message is well established (see Sect. 4).

We note that the choice to use AES-256 for both data encryption and QKD authentication is not just for the sake of simplicity, or so we can be confident our cryptosystem remains quantum-safe (although as this is our reason for using QKD in the first place, it is obviously important). Suppose that, despite all the analysis that has taken place up to this point, AES has an undisclosed flaw that allows attack 2 to be carried out by a select few. The result would be catastrophic. However, it would be no different compared to if the AES-based data encrypter had been paired with canonical BB84 instead, because the encryption can be broken directly in either case, meaning attack 2 offers no advantage. Of course, the chances of this happening are thought to be very low and so even if the one-time pad were used for data encryption, the comparative reduction in mathematical security is outweighed by increased resilience against DoS attacks.

In a world where Eve cannot compromise AES, she may carry out an unsuccessful version of attack 2 on only some of the qubits. Although Alice and Bob will be aware of her presence, there would be no way of knowing which qubits had been targeted in standard BB84, so the entire protocol would have to be aborted. In our case, the individual authentication of every basis would allow Alice and Bob to identify which qubits had been attacked in this way, giving them the option to keep those that were unaffected.

The above changes ensure that, if Eve tries to carry out attack 1, she will deny service for fractions of seconds rather than tens of minutes before her presence becomes obvious.

This is achieved without a reduction in the mathematical security of real-world QKD-based cryptosystems. The next step is to look at whether we can gain any further benefits by capitalising on our use of a computationally-secure MAC.

Now that every basis announcement is accompanied by an authentication tag, an interesting property emerges. There are only two possible tags for any given key/nonce pair, depending on whether the qubit was prepared in the X basis or the Z basis (though the exact values are unpredictable for anyone not in possession of the key). This means that if Alice decides to send the tags on their own, without the plaintext basis announcement that they authenticate, Bob can work out how he should have measured the qubit, by comparing the tags he would expect for each option.

Ideally, lack of knowledge about Alice and Bob's shared secret will prevent Eve from also identifying the correct bases using the authentication tags. That is, if they provide confidentiality, which is not a traditional requirement of a MAC, then sending the tags on their own means she will no longer be able to carry out photon number splitting (PNS) attacks on two-photon terms. This can easily be shown to be true for tags of the form given in equation (2), though we reserve a more complete discussion for Sect. 4.

From the above, we have established that transmitting the basis information as proposed means two-photon pulses can contribute to the secure key rate. However, it is still possible to implement an alternative method for PNS on higher-order multiphoton terms. All protocols are vulnerable to this unless, as in [15] and [16], decoy states are used. The strategy (which we call attack 3) can be described as follows. Eve performs a quantum nondemolition measurement on the number of photons in each pulse. She blocks all single and two-photon terms, but splits those containing three or more photons. She retains at least two photons in a quantum memory, and allows the remainder to carry on towards Bob. Eve then performs unambiguous state discrimination [17] on the qubits in her possession and returns a proportion of Alice's raw key dependent on the number of photons she split out of each pulse.

Of course, if the tags provide a level of confidentiality sufficient to prevent two-photon PNS, there is no longer any reason for them to be transmitted after Bob has measured the qubits, as Eve is unable to obtain the information required to perform a man-in-the-middle attack. If the tags are transmitted in advance, Bob can work out how he needs to measure before each qubit arrives, increasing the efficiency of the protocol from 50% to 100%.

Protocol 1 pulls together the methods we have developed for performing computationally-secure, but still quantum safe, QKD. A streamlined version is presented in Fig. 2, the details of which can be found in the next section. Up until now, we have focused solely on utilising AES, because of its ubiquity in modern communications, and position as the de facto quantum-safe alternative to the one-time pad. However, should AES ever become compromised in some way, it would be trivial to substitute in an alternative cipher (for example, the post-quantum security of Serpent-256 is currently under evaluation [18]).

While we have assumed the quantum key will be used in computationally-secure cryptosystems, it is still sensible to investigate the impact of a user who insists on encrypting their data with the one-time pad in a bespoke setting, despite its low efficiency and lack of authenticated encryption modes. In this scenario, we retain the advantages of our protocol but, as Sect. 4 will further dissect, also expect to acquire everlasting security [19] (the plaintext cannot be recovered from the information available to Eve if she develops

Protocol 1 BB84-AES (basic version)

SUMMARY: Alice expands a shared secret with Bob, using computationally-secure QKD and quantum-safe primitives.

1 *One-Time Setup.*
   (a) An l_k-bit secret is shared between Alice and Bob using out-of-band communications, a trusted third party or a post-quantum public-key algorithm.
   (b) An l_v-bit initialisation vector is transmitted from Alice to Bob in the clear, where $l_v \leq 64$.

2 *Nonce Generation.* A single-use number s_i is constructed by appending a $(128 - l_v)$-bit counter to the initialisation vector. The counter starts at 0 and increments after each call made to the generator. It must be maintained across all rounds of QKD that use the same initial shared secret, and is not to be confused with the index i used in the mathematics of this paper, where $1 \leq i \leq N$.

3 *Authentication Tags.*
   (a) The shared secret is split into a 256-bit cipher key, k_{C}, and an $(l_k - 256)$-bit hash key, k_{H}.
   (b) Alice generates a cryptographically-secure random bit, which is used to select a basis $b_i \in \{X, Z\}$, and computes the tag $\tau_i^{\mathrm{A}} = h_{k_{\mathrm{H}}}(b_i) \oplus \mathrm{AES}_{k_{\mathrm{C}}}(s_i)$. h is a universal hash function, the output of which can be called from memory after it has been evaluated once for each basis, and AES is the Advanced Encryption Standard block cipher.
   (c) Bob calculates $\tau_i^X = h_{k_{\mathrm{H}}}(X) \oplus \mathrm{AES}_{k_{\mathrm{C}}}(s_i)$ and $\tau_i^Z = h_{k_{\mathrm{H}}}(Z) \oplus \mathrm{AES}_{k_{\mathrm{C}}}(s_i)$.

4 *Key Exchange.*
   (a) Alice prepares a qubit $|\psi\rangle_i$ by generating a cryptographically-secure random bit, $B_i \in \{0, 1\}$, and encoding it in the basis b_i.
   (b) Alice sends τ_i^{A} to Bob, closely followed by $|\psi\rangle_i$.
   (c) Bob compares τ_i^{A} with τ_i^X and τ_i^Z, to identify the basis in which he should measure. Upon receipt of $|\psi\rangle_i$, he will return B_i with probability $100\% - q$, where q is the quantum bit error rate.
   (d) Bob announces whether or not the qubit arrived, by means of an authenticated response. He should maintain a separate nonce generator to Alice, paired with a different shared secret. As Bob's response need only be "Yes" or "No", he may choose to transmit it in the same way as Alice sends her bases.

5 *Loop.* Steps 3b, 3c and 4 are repeated for the remaining $N - i$ qubits sent from Alice to Bob. As multiple tags can be constructed in parallel, this may begin prior to completion of the previous iteration.

6 *Post-Processing.*
   (a) Error correction and privacy amplification are carried out as in BB84. The messages sent during this step can be authenticated in the same way as above.
   (b) l_k bits are taken from the final key and stored for use as the initial secret in the next round of QKD, and a new initialisation vector is publicly agreed upon.

unlimited computational power after key exchange is complete). This, along with perfect forward secrecy (previously generated keys will be unaffected if the initial shared secret has not been refreshed and the current round of the protocol becomes compromised),

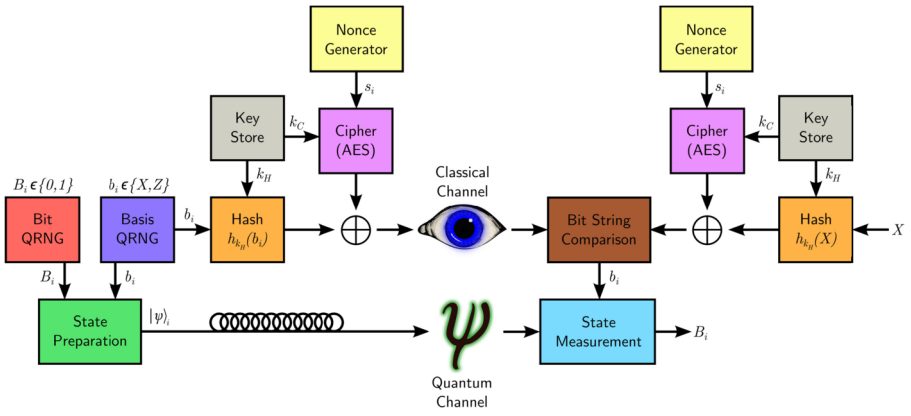

Figure 2 Block diagram showing the transmission of a single bit of key from Alice to Bob as part of BB84-AES in its reduced processing form

cannot be achieved if the key is encrypted directly with AES. For such a scheme, perfect forward secrecy is unattainable because anyone in possession of the long-term secret can use it to extract past session keys from the ciphertexts, rather than returning a set of bases that are no longer of any use. Similarly, compromising a previous shared secret at a later date will expose all keys distributed thereafter, even if the secret is updated after every key exchange with material from that session. Therefore, one should take care not to be fooled into thinking direct encryption of the key is a valid simplification of our protocol. Of course, a system based on this would not provide eavesdropper detection either.

4 Initial security analysis of BB84-AES

We now move to expand upon the claims of functionality made earlier in this paper. While we do not aim to provide a formal security proof for BB84-AES, there is a large body of literature that can be leveraged to perform an initial, high-level analysis. A more comprehensive proof should be the subject of further work, as the nature of our protocol means we will need to expand upon traditional approaches to assessing quantum security.

4.1 Rapid denial of service detection

If Eve tries to impersonate either Alice or Bob, the other party must be alerted to her presence by the authentication tag corresponding to the first qubit she sends after establishing a connection. The security of a MAC that accompanies a known message is well established when it takes the form of equation (2). For a 128-bit tag, all forgeries will be rejected with probability close to 1, so long as AES cannot be distinguished from a uniform random one-to-one function, an attacker sees no more than $\sqrt{\#\mathcal{K}_{\mathrm{M}}} = 2^{64}$ messages and, as in conventional QKD, our hash function has small differential probabilities [20]. Here, $\mathrm{AES}_{k_C}(s_i) \in \mathcal{K}_{\mathrm{M}}$ and $\#\mathcal{K}_{\mathrm{M}}$ represents the cardinality of the set. Of course, the protocol presented herein chooses to transmit the tags on their own rather than alongside a basis announcement, but the attacker gains no advantage from such a feature. The basis information can always be ignored, so the bound for rejecting forgeries will remain the same. As a result, just under 2^{64} bases can announced using a MAC, assuming Bob uses a separate initial secret key with an independent nonce for sending authenticated replies to Alice. For finite-key security, $\gtrsim 10^5$ raw bits must be exchanged and processed, meaning we can complete up to $\sim 10^{14}$ rounds of QKD before the scheme needs to be rekeyed.

The impact of this is two-fold. First, a key exhaustion attack is no longer viable, as an eavesdropper needs to establish more than eighteen billion connections before Alice and Bob will be prevented from constructing any more MACs of the form given by equation (2). Second, even if Eve were able to ensure key generation only failed at the very last moment, the number of times she would have to repeat her attack in order to exhaust Alice and Bob's shared secret is still on the order of a hundred trillion, given the rekeying limit specified above, and assuming they only began with the minimum number of bits required to construct a secure MAC. For networks of sufficient size, we would expect them to find a link that she cannot influence long before reaching that limit.

4.2 100% sifting efficiency

To avoid sifting the raw key, Bob must be able to obtain full information on the correct measurement bases from the authentication tags that Alice transmits in advance. Bob can only identify the correct basis so long as the MACs that represent each option are distinguishable from one another. Therefore, it is imperative that

$$h_{k_{\mathrm{H}}}(X) \neq h_{k_{\mathrm{H}}}(Z). \tag{3}$$

Consider a hash function family that is at least ε-almost universal, a condition fulfilled by those used in both of the MACs that we will recommend when considering how BB84-AES can be optimised [13, 14]. Then, the probability of violating equation (3) is

$$\mathrm{Prob(Collision)} \leq \varepsilon. \tag{4}$$

It is known that the MAC in which the hash family is used can be broken with success probability [21]

$$\mathrm{Prob(Successful\ attack)} \leq \varepsilon + \delta, \tag{5}$$

where δ is the chance of an attacker distinguishing AES from a truly random function, given that block ciphers can be considered pseudo-random functions (PRFs). Therefore,

$$\mathrm{Prob(Bob\ cannot\ obtain\ basis)} \leq \mathrm{Prob(Successful\ attack)}. \tag{6}$$

4.3 Authentication tag confidentiality

A radical difference between BB84-AES and all other forms of QKD is that we transmit the basis information ahead of the qubits. Therefore, the authentication tags must provide confidentiality against an eavesdropper, such that she cannot obtain any information on the correct measurement bases.

AES-CTR (AES running in Counter Mode [22]) encrypts an arbitrary message, m_j, as follows:

$$c_j = m_j \oplus \mathrm{AES}_{k_{\mathrm{C}}}(s_j), \tag{7}$$

where c_j is the ciphertext and s_j is a nonce. The security of Counter Mode with a PRF is discussed in [23], and this forms the foundation for showing that AES-CTR provides

confidentiality, by reason of block ciphers being considered strong pseudo-random permutations that can be treated as PRFs [24]. Up to 2^{64} messages can be encrypted with AES-CTR [23], so long as the counter contained within the nonce is of length 64 bits or more, with the remainder comprised of random bits. This limit is the same as that imposed previously to ensure unforgeability of the authentication tags.

Because AES-CTR is plaintext agnostic, it is perfectly legitimate to choose

$$m_j = h_{k_{\mathrm{H}}}(m_i), \tag{8}$$

where $h_{k_{\mathrm{H}}}(\cdot)$ is a keyed hash function, and m_i is also an arbitrary message. Therefore, equation (7) can be rewritten as

$$c_j = h_{k_{\mathrm{H}}}(m_i) \oplus \mathrm{AES}_{k_{\mathrm{C}}}(s_j). \tag{9}$$

We observe that when $s_j = s_i$ this is equivalent to equation (2), and so

$$\mathcal{T} \subset \mathcal{C}, \tag{10}$$

where $\mathcal{T}$ is the set of all possible authentication tags that take the form of equation (2) and $\mathcal{C}$ is the set of all possible ciphertexts that take the form of equation (7). Thus, our authentication tags provide confidentiality with regards to the output of the hash function, assuming that AES is quantum-safe.

4.4 Resistance to photon number splitting attacks on two-photon pulses

We can demonstrate two-photon PNS resistance by considering whether an eavesdropper is able to obtain more information on the final key through a two-photon number splitting attack than if SARG04 were to be used instead. In SARG04, Alice publicly declares two possibilities for the state she transmitted, instead of announcing the basis she prepared in. If Eve wants to obtain full information on the key by taking advantage of multi-photon terms, she must carry out attack 3, blocking all pulses containing less than three photons and performing unambiguous state discrimination on the remainder [17, 25].

The confidentiality provided by our authentication tags is, from an attacker's perspective, equivalent to Alice not announcing the bases at all. We could choose to announce two possible states as in SARG04, and then the attacker would have the same amount of information on the final key. Not making this announcement gives the attacker zero advantage, as they can always discard the information if it is given to them. Therefore, BB84-AES is at least as resilient as SARG04 against PNS attacks on two-photon pulses, so long as AES remains secure.

4.5 Perfect forward secrecy when combining BB84-AES with encryption based on the advanced encryption standard block cipher

In order for our protocol to have perfect forward secrecy, an attacker who compromises the initial shared secret during one round of the protocol must not be able to use this to obtain keys that were distributed using the same initial shared secret in previous rounds of the protocol. An attacker who compromises the initial shared secret from a previous round gains the ability to forge tags from that round (though to no effect as key exchange

is already complete) and find out the bases used. This is also the case if an attacker gains unlimited computational power. Therefore, if we can prove everlasting security of BB84-AES when encrypting data with the OTP, perfect forward secrecy will follow.

4.6 Everlasting security when combining BB84-AES with the one-time pad encryption scheme

An attacker who gains unlimited computational power after the conclusion of the protocol must not be able to gain any knowledge on the key from the information transmitted in the authentication tags, assuming AES remained secure for the duration of the protocol. In BB84-AES, the authentication tags are used to secretly communicate a subset $\mathcal{I}$ of the classical information exchanged by Alice and Bob. In standard BB84, $\mathcal{I}$ is communicated publicly during the protocol, after all qubits have been exchanged. This means that after the conclusion of BB84, $\mathcal{I}$ is known to the attacker, and the fact this does not compromise the security is of fundamental importance in QKD [26]. Therefore, if an attacker manages to extract $\mathcal{I}$ after the conclusion of BB84-AES, the protocol remains secure, as they have no more information than in the standard case.

However, an attacker who gains unlimited computational power after the conclusion of the protocol must also not be able to gain any knowledge on the key by exploiting the newly forgeable authentication tags, assuming AES remained secure for the duration of the protocol. In [19], it is shown that, for computationally-secure QKD, bounds on the attacker's classical runtime, quantum runtime and quantum memory need only be applied to ensure the classical channel cannot be tampered with during the course of the protocol. Afterwards, standard QKD arguments hold, whereby the authenticity of the classical channel is no longer of relevance, even in the case of general attacks.

As we are considering an attacker who cannot inject, reorder or modify authentication tags that were sent and received in the past, we would expect BB84-AES to have everlasting security when used with the OTP, so long as Eve was sufficiently bounded during the execution of the protocol such that she was unable to break the computationally-secure authentication scheme. For security against quantum computers, this means we are assuming AES is a quantum PRF, although there is no guarantee this will follow from the fact that block ciphers may be considered standard PRFs [27].

4.7 The role of randomness in BB84-AES

Finally, we will show that in the absence of an attacker, keys output by BB84 and BB84-AES are equally random. Since the authentication tags are used only in the communication of information, this boils down to asking whether Bob's failure to inject additional random numbers has an adverse effect on the entropy of the final key. The short answer is no, and it is important to realise that any answer to the contrary would also apply in the case where Alice and Bob both randomly generate the same set of bases with probability $\frac{1}{2^N}$. If Alice is using an ideal quantum random number generator (QRNG) then the key she transmits will have maximum entropy. In conventional QKD, Bob's random bit deletion becomes a matter of practicality rather than doing anything to further mitigate Eve's ability to guess the final key, assuming he also uses an ideal QRNG. Therefore, removing this step does nothing to reduce the randomness in the output of BB84-AES.

However, the situation changes somewhat if an insecure or backdoored random number generator (RNG) is used for basis selection at either end. While the outcome is trivial

when the same RNG is used for Alice's bit selection (an eavesdropper will be able to obtain the key without further interference), this is not enforced, so we stick to a more general implementation where different RNGs are used for Alice's bits, Alice's bases and Bob's bases. This configuration gives rise to two possible attacks in standard BB84. If Eve can anticipate Alice's random sequence, she will be able to intercept the qubits, measure in the correct basis and resend. Assuming zero errors, her measurements return the same raw key as Alice, which can be correctly sifted when the bases are publicly compared (attack 4). Similarly, if Eve can anticipate Bob's random sequence, she will be able to intercept the qubits, measure using his set of bases and resend. Assuming zero errors, her measurements return the same raw key as Bob which can be correctly sifted when the bases are publicly compared (attack 5).

In BB84-AES, attack 5 reduces to attack 4 without sifting. As Bob is not generating any extra randomness himself, the predictability of his measurement bases is determined by Alice's RNG. Therefore, Bob needs to trust Alice has made sensible implementation decisions but, given attack 4 exists in conventional QKD anyway, this is nothing new and Eve's ability to exploit a faulty RNG remains unaffected.

5 Comparing BB84-AES with the state of the art

As BB84-AES is 100% efficient, it does not need to be implemented using biased bases which, conditional on the number of photons transmitted, are used to asymptotically double the efficiency of BB84 [28]. In fact, given we have already waived our interest in information-theoretic security, transmitting the tags in advance of the qubits is a slightly preferable solution. This is partly because the efficiencies of real and simulated biased basis experiments are still noticeably lower than 100% [29, 30], however assuming no additional countermeasures are employed, the protocol described in [28] is also vulnerable to a more simplistic PNS attack than that which is applicable to vanilla BB84. Assume Eve does not possess a quantum memory, but is otherwise unchanged. She performs a quantum nondemolition measurement on the number of photons in each pulse and blocks all single-photon terms. For the remainder, she splits off at least one photon from every pulse, and allows at least one photon to carry on towards Bob. Eve immediately measures her copy in the key generation basis. When Alice and Bob publicly sift their qubits, she can identify those used for eavesdropper detection, and discard any information she has on them. Every bit of her final key has now been correctly measured, without revealing her presence. This is possible due to the recommendation that key be generated from a single basis, with the other used only for eavesdropper detection. The fact a quantum memory is no longer required makes it a much more realistic exploit for modern-day implementations than standard PNS attacks, emphasising why it is imperative to use decoy states in any current system relying on biased bases. In contrast, the aforementioned *Clavis*^{2} predominantly uses unbiased SARG04 [31], which has the same level of PNS-resistance as the protocol described herein, and falls back on unbiased BB84 for short distances, where SARG04 is not proven secure [25]. This may be considered acceptable so long as quantum memories remain in the early stages of development.

Table 1 presents a comparison between BB84-AES, biased basis BB84, SARG04 and decoy state BB84. In the case of decoy state protocols, it is especially important to highlight that hardware changes are required in order to upgrade systems which are installed on contemporary QKD networks and do not use decoy states. Although we have already

Table 1 Comparing BB84-AES, BB84 with biased bases, SARG04 and BB84 with decoy states. It is possible to combine biased basis and decoy state BB84, with sifting efficiency $\lim_{N\to\infty} \zeta = \text{Prob(Signal)}$

	BB84-AES	Biased Basis BB84	SARG04	Decoy State BB84
Mathematical Security with One-Time Pad	PC (short term) IT (long term)†	IT	– (low loss) IT (higher loss)	IT
Mathematical Security with AES-GCM	PC	PC	– (low loss) PC (higher loss)	PC
Endpoint Denial of Service Resistance	Yes	No	No	No
Photon Number Splitting Resistance	Two-photon	No	Two-photon	Yes
Sifting Efficiency (ζ)	100%	$\lim_{N\to\infty} \zeta = 100\%$	25%	$\frac{1}{2} \times$ Prob(Signal)
Requires Hardware Changes	No	Sometimes	No	Yes

† Here, long-term security works under the assumption that the scheme was not broken at the time of key exchange.
Key: IT = Information Theoretic; PC = Practical Computational; – = Unproven.

stated that systems such as the *Clavis*[2] can continue to be used at present, some network operators may find that the financial cost to eventually replace these is too high to justify. Therefore, a software patch allowing them to run BB84-AES may be a more practical solution.

We also note that BB84-AES and decoy state QKD are not necessarily mutually exclusive. Under normal operation, decoy state BB84-AES would allow users to benefit from the DoS resistance and efficiency of BB84-AES, along with the enhanced PNS resistance of decoy states, which may be of particular interest when dealing with an eavesdropper who is able to carry out attack 3. If a modulator were to break on an Alice unit, it may then be possible to continue performing QKD without decoy states until the system can be repaired or replaced, if it is acceptable to only maintain two-photon PNS resistance in the interim.

However, if Alice and Bob do not wish to undergo an extra round of public announcements to identify which of Bob's detections were decoy states, then Alice must transmit them in the same way as she communicates the bases. She may send an extra tag with each qubit, but this will double the bandwidth requirements of the protocol, meaning it would make more sense to include decoy state options for each authentication tag, in addition to the standard X and Z basis choices. This will cause complications for some of the optimisations we present in the following section, so further work is required to establish the most efficient way in which decoy states can be used with BB84-AES.

6 Optimisations

While it is perfectly feasible to implement Protocol 1 as presented herein, there are a number of changes that can be made to reduce demand on the computational and/or communications resources. The first of these is summarised in Protocol 2, where we allow Bob to check only whether the tag he receives is a match for that corresponding to a measurement in the X basis. This requires marginally less memory and processing time than individual basis authentication in otherwise-standard BB84. The trade-off is that if Eve measures in the Z basis, she no longer needs to be able to forge the corresponding authentication tag,

Protocol 2 BB84-AES (reduced processing)

SUMMARY: Replaces steps 3c and 4c in Protocol 1, halving the number of XOR operations and tag comparisons that Bob has to carry out.

3 *Authentication Tags.*
   (c) Bob calculates $\tau_i^X = h_{k_H}(X) \oplus \mathrm{AES}_{k_C}(s_i)$.

4 *Key Exchange.*
   (c) Bob compares τ_i^A with τ_i^X. If it matches, he will choose to measure in the X basis. Otherwise, he will choose to measure in the Z basis. Upon receipt of $|\psi\rangle_i$, he will return B_i with probability $100\% - q$, where q is the quantum bit error rate.

Can be combined with: BB84-AES (reduced bandwidth).

Table 2 Showing the probability of a bit-flip error occurring between Alice and Bob depending both on the bases chosen by each of the three parties, and whether or not Eve blindly modifies the authentication tag

Alice's Basis	Eve's Basis	Forwarding Choice	Bob's Basis	Prob (error)
X	X	$\tau_i^E = \tau_i^A$	X	0
X	X	$\tau_i^E \neq \tau_i^A$	Z	0.5
X	Z	$\tau_i^E = \tau_i^A$	X	0.5
X	Z	$\tau_i^E \neq \tau_i^A$	Z	0.5
Z	X	$\tau_i^E = \tau_i^A$	Z	0.5
Z	X	$\tau_i^E \neq \tau_i^A$	Z	0.5
Z	Z	$\tau_i^E = \tau_i^A$	Z	0
Z	Z	$\tau_i^E \neq \tau_i^A$	Z	0

ensuring only that the one she forwards, τ_i^E, is different to that sent by Alice. However, Eve still has not broken the authentication scheme (she cannot obtain any basis information or force Bob to measure in the X basis), and so this kind of interference will be exposed by the quantum bit error rate (QBER). Table 2 gives the outcomes for all of Eve's possible strategies. It is clear that $\tau_i^E \equiv \tau_i^A$ remains optimal.

Next, we look at the effect of requiring the classical channel to transmit 128× the number of bits transferred over the quantum channel. Given the *Clavis*^{2} emits laser pulses clocked at 5 MHz [32], the classical data rate needs to be 640 Mbit/s. For comparison, the Bristol and UK quantum networks on which the *Clavis*^{2} systems are being deployed, both have SFP+ and QSFP+ channels with capacities of 10 Gbit/s and 40 Gbit/s respectively. While the gap appears large between what we need and what we can provide, pre-commercial quantum hardware has been shown to be capable of reaching super-GHz clock speeds [33]. Due to the way in which the states were encoded in this example, the actual clock rate of BB84 was only 560 MHz, however to avoid a potential future where our protocol necessitates two transceivers be multiplexed together, we can reduce our tag lengths as described in Protocol 3. This remains secure for up to 2^{32} messages [20], allowing $\sim 10^4$ full rounds of QKD per initial key, and brings the classical communications requirements to within the capabilities of QSFP28 or CFP4 transceivers.

The final optimisation reduces demand on the classical channel by grouping multiple bases into a single authentication tag (Protocol 4). The time taken to establish the presence of a fake user should not change significantly, because the tags are still transmitted ahead of the first qubit in every group. Of course, the processing at Bob's end will be expected to take slightly longer than before, as a MAC that represents ξ bases will have $\beta = 2^\xi$

Protocol 3 BB84-AES (reduced bandwidth)

SUMMARY: Replaces the 128-bit tags in Protocol 1 with 64-bit tags of the same form. UMAC [13] and VMAC [14] both provide such functionality, without dropping below the required security level.

Can be combined with: BB84-AES (reduced processing), BB84-AES (dense information transfer).

Protocol 4 BB84-AES (dense information transfer)

SUMMARY: Replaces steps 3b, 3c, 4a, 4b, 4c and 5 in Protocol 1, grouping multiple bases into a single tag to reduce the necessary channel capacity by a factor of $l_\tau(\xi-1)$. l_τ is the tag length in bits, and ξ is the number of bases per tag. We redefine the range of i such that $1 \leq i \leq \frac{N}{\xi}$.

3 *Authentication Tags.*
   (b) Alice generates ξ cryptographically-secure random bits, which are used to select bases b_η through $b_{\eta+\xi-1}$, where $b_{\eta+\Xi} \in \{X, Z\}$, $\eta = 1 + (i-1)\xi$ and $\Xi \in \{0, \ldots, \xi-1\}$. It is required that $1 < \xi \ll N$. She computes the tag $\tau_i^{\mathrm{A}} = h_{k_{\mathrm{H}}}(b_\eta || \ldots || b_{\eta+\xi-1}) \oplus \mathrm{AES}_{k_{\mathrm{C}}}(s_i)$. h is a universal hash function, AES is the Advanced Encryption Standard block cipher, and $||$ is used to indicate a concatenation.
   (c) Bob calculates $h_{k_{\mathrm{H}}}(b_\eta || \ldots || b_{\eta+\xi-1})$ for all 2^ξ possible values of $b_\eta || \ldots || b_{\eta+\xi-1}$, storing the results in ascending order. He also evaluates $\mathrm{AES}_{k_{\mathrm{C}}}(s_i)$ separately.

4 *Key Exchange.*
   (a) Alice prepares the qubits $|\psi\rangle_\eta$ to $|\psi\rangle_{\eta+\xi-1}$. This is done by generating ξ cryptographically secure random numbers B_η through $B_{\eta+\xi-1}$, where $B_{\eta+\Xi} \in \{0, 1\}$, and encoding them in the bases b_η through $b_{\eta+\xi-1}$ respectively.
   (b) Alice sends τ_i^{A} to Bob, closely followed by all $|\psi\rangle_{\eta+\Xi}$ for the corresponding value of i.
   (c) Bob computes $\tau_i^{\mathrm{A}} \oplus \mathrm{AES}_{k_{\mathrm{C}}}(s_i)$ and checks it against the lookup table he constructed in step 3c, to identify the bases in which he should measure. Upon receipt of $|\psi\rangle_{\eta+\Xi}$, he will return $B_{\eta+\Xi}$ with probability $100\% - q$, where q is the quantum bit error rate.

5 *Loop.* Steps 3b, 3c and 4 are repeated for the remaining $N - i\xi$ qubits sent from Alice to Bob. As multiple tags can be constructed in parallel, this may begin prior to completion of the previous iteration.

Can be combined with: BB84-AES (reduced bandwidth).

possible values for each key/nonce pair. His method for identifying the correct set of measurements differs from Protocol 1 in that he must compute all possible hashes and store them in a lookup table. He can then XOR the incoming tag with the AES-generated key, and compare. Combining Protocol 3 with Protocol 4 will speed up the hash function [14], thereby reducing the time taken to construct the table. The necessary calculations can be performed during downtime, or in parallel with device and fibre characterisation, or in parallel with a previous round of QKD provided each initial shared secret is used across multiple rounds. An important subtlety, that is also true for Protocols 1, 2 and 3, is the

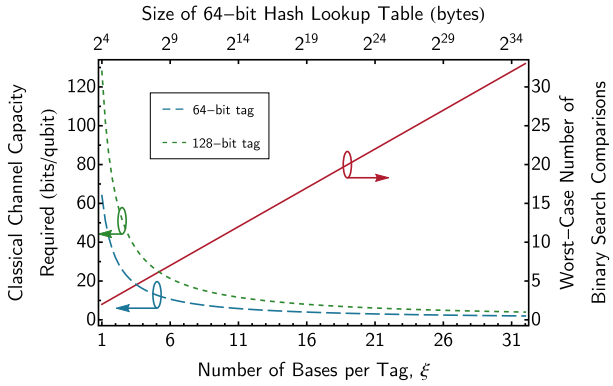

Figure 3 Illustrating how changing the number of bases represented by a single authentication tag affects both classical communication and computational resource requirements. To get a rough estimate for how our protocol will perform on a particular physical system, one can multiply the classical channel capacity by the QKD clock rate, and the worst-case number of comparisons by the time taken to perform a single binary search comparison. The size of a 128-bit hash lookup table will always be double that of its 64-bit counterpart

hashes only need to be computed once so long as the initial secret remains unchanged, meaning that until this is refreshed, the lookup table does not need to be reconstructed.

To prevent a simple timing attack, Alice can never send the qubits until the worst-case lookup time has elapsed, so Bob must take care to select a search algorithm that is optimal in this regard, such as binary search [34] which makes no more than $\lfloor \log_2 \beta \rfloor + 1 = \xi + 1$ comparisons.

The exact value of ξ reflects a trade-off between computational and communications resources, and it is clear from Fig. 3 that the greatest benefits can be achieved when $1 < \xi \ll 32$, because of the exponential behaviour of both classical channel capacity and memory requirements. As a concrete example, we will consider the Bristol Quantum Network, which is hosted on pre-existing infrastructure, with each node's server containing 64 Intel Xeon E5-2697A v4 processors. By implementing a binary search on a single CPU, without hardware-specific optimisation, we can estimate the performance of our protocol on a real system. If we assume a 64-bit tag and want to employ only a single SFP+ (QSFP+) channel, then $\xi = 8$ ($\xi = 2$) maximises the QKD clock rate while trying to use the least possible memory. In this case, it takes 6.940 ± 0.085 ns (2.085 ± 0.017 ns) to run the search, allowing for a 1.153 ± 0.014 GHz (0.959 ± 0.008 GHz) clock and consuming 2048 bytes (32 bytes) of memory, out of 87.7 GiB available and 131.7 GiB total RAM. To run a hypothetical 1.72 GHz-clock BB84 device based on the technology in [33] would require $\xi = 12$ ($\xi = 3$). In this instance, the search takes 9.692 ± 0.039 ns (2.881 ± 0.036 ns), and 32,768 bytes (64 bytes) of memory is required. However, it is important to note that while these parameters are sufficient to enable the use of presently-installed transceivers, the quantum clock is still capped at 1.238 ± 0.005 GHz (1.041 ± 0.013 GHz) because of the maximum search time. Hence, some parallelisation will also be required, in that each search must begin before the previous one is guaranteed to have finished, should the quantum clock need to exceed this limit. Of course, our use of computationally-secure data encryption means our secret key rate and, by extension, our quantum clock rate, does not need to be as high as if the one-time pad were being used instead. For AES-GCM, $2^{39} - 256$ bits of information can be encrypted with every 256-bit key [35], so our secret key rate can be on the order of 10^{-10} times that which would be necessary to encrypt the same amount

of data with the one-time pad. As a result, it has been shown separately that the classical bandwidth requirements of our protocol will be $\ll 1\%$ of the total channel capacity [10].

Technically, the higher the value of ξ, the easier it is for Eve to guess one of the $2^{\xi} - 1$ other authentication tags that Bob will accept. A correct guess is still highly improbable, and so she will almost certainly be detected, however even if successful, Eve controls only whether or not Bob measures with the same bases as Alice. Hence, this is nothing more than a restricted version of the strategy she can employ in Protocol 2 and, in the unlikely case of an odds-defying set of forgeries, Alice and Bob will be made aware of Eve's presence by the QBER.

Similarly, the chance of a collision between different entries in the lookup table also increases with ξ. This probability will remain small so long as a suitable hash function is chosen, and the analysis in Sect. 4.2 is still valid, however it would be prudent to include a collision checker in any practical implementation.

7 Analysis

The advantages of BB84-AES are possible only so long as the output of the cipher used to construct our authenticators is indistinguishable from the output of a random permutation. This criterion is the same as that for ensuring the security of quantum-safe encryption schemes used in day-to-day communications, so having to sacrifice information-theoretic security is not overly concerning. At any rate, the chance that the above assumption will be violated is far lower than the likelihood of an attacker exploiting one of the weaknesses that our protocol defends against. If one were to insist on unconditional security, individual basis authentication could be performed using AES tags in standard BB84, reauthenticating everything at the end with a traditional Wegman–Carter MAC. However, attack vectors may still exist for exhausting the initial shared secret and, given the issues we have raised over implementing biased bases without the necessary hardware for decoy states, BB84-AES remains preferable, particularly for minimalistic implementations and retrofitting systems already in the field.

One may even wish to go a step further with regards to modifying the classical channel, because authenticated modes of encryption have the same properties as our authentication tags. Throughout this work, we have assumed AES-GCM is being used to protect our data, and the question arises as to what happens when QKD incorporates such a scheme in its entirety, rather than just capitalising on the block cipher.

BB84-A/G, which supplants the computationally-secure MAC with AES-GCM, should behave in much the same way as BB84-AES, with one important difference. As all of AES-GCM's possible failure criteria are now contained within those for BB84-A/G, the maximum failure probability of the overall system can be defined entirely by the maximum failure probability of BB84-A/G.

This can be expressed mathematically as follows. The ε-security of a confidential cryptosystem that is built from independent and composable subsystems is quantified using [36]

$$\varepsilon_{\text{total}} \leq \varepsilon_{\text{dist}} + \varepsilon_{\text{enc}}. \tag{11}$$

Here, $\varepsilon_{\text{dist}}$ is the deviation from perfection of a key distribution protocol and its output, while ε_{enc} is the same metric, applied to the authenticated data encryption instead.

The composability of BB84-AES is not guaranteed, emphasising the need for a full security proof. Nonetheless, if it does possess this essential property, $\varepsilon_{\text{total}}$ for BB84-AES with AES-GCM encryption will be calculable from equation (11). In contrast, AES-GCM never fails on its own when used with BB84-A/G, so we can apply the following:

$$\varepsilon_{\text{total}} = \text{Max}(\varepsilon_{\text{dist}}, \varepsilon_{\text{enc}}) = \varepsilon_{\text{dist}}. \tag{12}$$

This comes with one important caveat. As soon as we consider applications beyond AES-GCM or AES-CTR, equation (12) no longer applies. Therefore, if BB84-A/G is to be used in an arbitrary cryptosystem, its security should be evaluated under the expectation that the operation in which the key will be used is completely independent.

Adapting our work for BBM92 [37] (which we call BBM92-AES) and the Six State Protocol [38] (likewise, SSP-AES) is trivial. In the case of the former, the public channel is identical to that of BB84. For the latter, we must compute an extra tag, which we define to be

$$\tau_i^Y = h_{k_{\text{H}}}(Y) \oplus \text{AES}_{k_{\text{C}}}(s_i). \tag{13}$$

Consequently, a reduced processing variant would need to test authentication tags corresponding to two out of three bases (cf. Protocol 2). Like with the six-state version of SARG04 [39], we expect Eve's attacks on multi-photon terms to be further restricted, such that she can only perform unambiguous state discrimination on weak coherent pulses that contain at least five photons. This is because, given an r-photon pulse, the upper bound on the number of states that Eve can discriminate between is $r + 1$ [40].

If we consider the commonly-chosen mean photon number $\mu = 0.1$, then the probability of generating a pulse containing five or more photons is

$$\begin{aligned} \text{Prob}(n \geq 5) &= 1 - \sum_{r=0}^{4} \text{Prob}(n = r) \\ &= 1 - e^{-\mu} \sum_{r=0}^{4} \frac{\mu^r}{r!} \\ &= 7.67 \times 10^{-8}. \end{aligned} \tag{14}$$

As a result, we roughly expect to see a five-photon term only once every 130 keys if the protocol concludes immediately upon reaching the finite key limit ($\sim 10^5$ bits). However, if we now consider $\mu = 0.5$, which is the optimal mean photon number for decoy state QKD [15], then

$$\text{Prob}(n \geq 5) = 1.74 \times 10^{-4}. \tag{15}$$

Here, several tens of attackable pulses will be transmitted per key. We would expect Alice and Bob to notice the cataclysmic drop in rates if Eve were to block all but these. Yet there may still be attack strategies that allow her to gain useful information by performing unambiguous state discrimination on a fraction of the key, hence the need for a more thorough investigation into the potential role of decoy states in SSP-AES.

Instead of just considering the impact of applying our authentication tags to other QKD protocols, we may also wish to ask what happens if they are used elsewhere in BB84-AES. Can any advantage be gained if Eve does not know which qubits arrived, because Bob notifies Alice in the same way as she informs him of the correct bases? And what is the effect of using the authentication tags to encrypt error correction parities in CASCADE? This last question is similar to a situation that has previously been considered, in which the parities are encrypted using a OTP as a way of guaranteeing information-theoretic security [41, 42]. Here, the obvious downside is that the number of parity checks must be taken into account when calculating the secret key rate [43]. However, extending our authentication tags to the error correction stage would use no additional key, so while the security implications would need to be thoroughly examined, this may be of benefit.

A final novelty of our protocol is that, by daisy-chaining multiple Alice/Bob pairs, it is possible to supply an arbitrary amount of quantum-safe quantum randomness with everlasting security to someone who cannot directly access a node containing a QRNG. Of course, the resource requirements scale badly (for a chain of d nodes, the QRNG would need to generate 2^{d-1} bit strings) and while the idea may be academically interesting, it is unclear whether such functionality is of any real-world use.

8 Conclusion

We have shown that, by reducing the mathematical security of BB84, it is possible to almost instantly detect denial of service that leverages fake users, something which no other quantum key distribution protocol has been shown to be capable of. Our design is inherently resilient against attacks that aim to exhaust Alice and Bob's supply of initial secret key, but does not lead to large memory overheads because of this (to achieve the same performance in this regard as the basic version of our protocol, vanilla BB84 would require petabytes of initial shared secret), nor does it operate reactively by falling back on public-key cryptography. In changing how and when the bases are announced, we are able to achieve exactly 100% efficiency and, instead of posing a risk to security, two-photon terms now contribute positively to the final key rate, independently of the distance or number of bits exchanged, and without any further cost. As the quantum channel is the same as that used for BB84, the simplicity of state preparation is retained and the amount of loss that can be tolerated will be unchanged.

In developing BB84-AES, and showing the practical benefits it can provide, we have shown that the intersection between modern and quantum cryptography should be explored in more detail, with greater collaboration between researchers on both sides, as this area still seems largely untapped and ripe for real-world improvements in algorithms and implementations.

Acknowledgements
Thanks go to K.G. Paterson and D.L.D. Lowndes for useful conversations.

Funding
ABP was supported by the Bristol Quantum Engineering Centre for Doctoral Training, EPSRC grant EP/L015730/1. The authors also acknowledge the UK Quantum Technology Hub for Quantum Communications Technologies, EPSRC grant EP/M013472/1.

Availability of data and materials
The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Competing interests
While the work was being carried out and a first draft of this paper was written, the authors had no competing interests. Since then, CE has moved to become the CEO of a quantum security company, focusing on the commercialisation of integrated QKD hardware.

Authors' contributions
ABP identified the denial of service attack and developed the BB84-AES protocol along with its variants. ABP also carried out the initial, high-level analysis of security and functionality. The work was supervised by JGR and CE who examined the protocol for errors and put forward possible simplifications. All authors read and approved the final manuscript.

Publisher's Note

Received: 15 July 2019 Accepted: 22 April 2020 Published online: 01 May 2020

References

1. Bennett CH, Brassard G. Quantum cryptography: public key distribution and coin tossing. In: Proceedings of IEEE international conference on computers, systems and signal processing. vol. 1. 1984. p. 175–9.
2. Gottesman D, Lo H-K, Lütkenhaus N, Preskill J. Security of quantum key distribution with imperfect devices. Quantum Inf Comput. 2004;4(5):325–60.
3. Ben-Or M, Horodecki M, Leung DW, Mayers D, Oppenheim J. The universal composable security of quantum key distribution. In: Kilian J, editor. TCC 2005: theory of cryptography. Lecture notes in computer science. vol. 3378. 2005. p. 386–406.
4. CESG: quantum key distribution. *White Paper* 2016.
5. Wegman MN, Carter JL. New hash functions and their use in authentication and set equality. J Comput Syst Sci. 1981;22:265–79.
6. Stucki D, Legré M, Buntschu F, Clausen B, Felber N, Gisin N, Henzen L, Junod P, Litzistorf G, Monbaron P, Monat L, Page J-B, Perroud D, Ribordy G, Rochas A, Robyr S, Tavares J, Thew R, Trinkler P, Ventura S, Voirol R, Walenta N, Zbinden H. Long-term performance of the SwissQuantum quantum key distribution network in a field environment. New J Phys. 2011;13(12):123001.
7. NIST: specification for the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication. 2001.
8. Sasaki M, Fujiwara M, Ishizuka H, Klaus W, Wakui K, Takeoka M, Miki S, Yamashita T, Wang Z, Tanaka A, Yoshino K, Nambu Y, Takahashi S, Tajima A, Tomita A, Domeki T, Hasegawa T, Sakai Y, Kobayashi H, Asai T, Shimizu K, Tokura T, Tsurumaru T, Matsui M, Honjo T, Tamaki K, Takesue H, Tokura Y, Dynes JF, Dixon AR, Sharpe AW, Yuan ZL, Shields AJ, Uchikoga S, Legré M, Robyr S, Trinkler P, Monat L, Page J-B, Ribordy G, Poppe A, Allacher A, Maurhart O, Länger T, Peev M, Zeilinger A. Field test of quantum key distribution in the Tokyo QKD network. Opt Express. 2011;19(11):10387–409.
9. Roscino R, Layat K, Ribordy G, Huttner B, Caselunghe D. Applicability of a post-quantum signature in a QKD public channel (abstract). 6th International Conference on Quantum Cryptography (QCRYPT). 2016.
10. Price AB. Pragmatic quantum cryptography in next-generation photonic networks. PhD thesis. University of Bristol. 2019.
11. Brassard G. On computationally secure authentication tags requiring short secret shared keys. In: Chaum D, Rivest RL, Sherman AT, editors. Advances in cryptology: proceedings of crypto. vol. 82. 1983. p. 79–86.
12. Bernstein DJ. The Poly1305-AES message-authentication code. In: Gilbert H, Handschuh H, editors. Fast software encryption. FSE 2005. Lecture notes in computer science. vol. 3557. 2005. p. 32–49.
13. Black J, Halevi S, Hevia A, Krawczyk H, Krovetz T, editors. UMAC: message authentication code using universal hashing. Network Working Group, The Internet Society. 2006.
14. Krovetz T. Message authentication on 64-bit architectures. In: Biham E, Youssef AM, editors. Selected areas in cryptography: 13th international workshop. SAC 2006 revised selected papers. Lecture notes in computer science. vol. 4356. 2007. p. 327–41.
15. Lo H-K, Ma X, Chen K. Decoy state quantum key distribution. Phys Rev Lett. 2005;94(23):230504.
16. Stucki D, Brunner N, Gisin N, Scarani V, Zbinden H. Fast and simple one-way quantum key distribution. Appl Phys Lett. 2005;87(19):194108.
17. van Enk SJ. Unambiguous state discrimination of coherent states with linear optics: application to quantum cryptography. Phys Rev A. 2002;66(4):042313.
18. Augot D, Batina L, Bernstein DJ, Bos J, Buchmann J, Castryck W, Dunkelman O, Güneysu T, Gueron S, Hülsing A, Lange T, Mohamed MSE, Rechberger C, Schwabe P, Sendrier N, Vercauteren F, Yang B-Y. Initial recommendations of long-term secure post-quantum systems. PQCRYPTO. 2015.
19. Mosca M, Stebila D, Ustaoğlu B. Quantum key distribution in the classical authenticated key exchange framework. In: Gaborit P, editor. Post-quantum cryptography. PQCrypto 2013. Lecture notes in computer science. vol. 7932. 2013. p. 136–54.
20. Bernstein DJ. Stronger security bounds for Wegman–Carter–Shoup authenticators. In: Advances in cryptology: EUROCRYPT 2005. Lecture notes in computer science. vol. 3494. 2005. p. 164–80.
21. Krovetz TD. Software-optimized universal hashing and message authentication. PhD thesis, University of California Davis. 2000.
22. Dworkin M. NIST SP 800-38A: recommendation for block cipher modes of operation. National Institute of Standards and Technology. 2001.
23. Bellare M, Desai A, Jokipii E, Rogaway P. A concrete security treatment of symmetric encryption. In: Proceedings of the 38th symposium on foundations of computer science. 1997. p. 394–403. Full version available from http://web.cs.ucdavis.edu/~rogaway/papers/sym-enc.pdf [Last Accessed: 07/04/17].
24. Katz J, Lindell Y. Introduction to modern cryptography. London: Chapman & Hall; 2008.

25. Branciard C, Gisin N, Kraus B, Scarani V. Security of two quantum cryptography protocols using the same four qubit states. Phys Rev A. 2005;72(3):032301.
26. Shor PW, Preskill J. Simple proof of security of the BB84 quantum key distribution protocol. Phys Rev Lett. 2000;85(2):441–4.
27. Zhandry M. How to construct quantum random functions. In: Proceedings of the 53rd symposium on foundations of computer science. 2012. p. 679–87.
28. Lo H-K, Chau HF, Ardehali M. Efficient quantum key distribution scheme and a proof of its unconditional security. J Cryptol. 2005;18(2):133–65.
29. Erven C, Ma X, Laflamme R, Weihs G. Entangled quantum key distribution with a biased basis choice. New J Phys. 2009;11(4):045025.
30. Wei Z, Wang W, Zhang Z, Gao M, Ma Z, Ma X. Decoy-state quantum key distribution with biased basis choice. Sci Rep. 2013;3:2453.
31. Scarani V, Acín A, Ribordy G, Gisin N. Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations. Phys Rev Lett. 2004;92(5):057901.
32. ID Quantique SA: quantum key distribution system Clavis2 user guide (v 3.0). 2013.
33. Sibson P, Erven C, Godfrey M, Miki S, Yamashita T, Fujiwara M, Sasaki M, Terai H, Tanner MG, Natarajan CM, Hadfield RH, O'Brien JL, Thompson MG. Chip-based quantum key distribution. Nat Commun. 2017;8:13984.
34. Knuth DE. The art of computer programming. 2nd ed. vol. 3. Reading: Addison-Wesley; 1998.
35. Dworkin M. NIST SP 800-38D. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. National Institute of Standards and Technology. 2007.
36. Müller-Quade J, Renner R. Composability in quantum cryptography. New J Phys. 2009;11(8):085006.
37. Bennett CH, Brassard G, Mermin ND. Quantum cryptography without Bell's theorem. Phys Rev Lett. 1992;68(5):557–9.
38. Bechmann-Pasquinucci H, Gisin N. Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography. Phys Rev A. 1999;59(6):4238–48.
39. Tamaki K, Lo H-K. Unconditional secure key distillation from multi-photons. Phys. Rev. A 2006.
40. Chefles A. Unambiguous discrimination between linearly dependent states with multiple copies. Phys Rev A. 2001;64(6):062305.
41. Lütkenhaus N. Estimates for practical quantum cryptography. Phys Rev A. 1999;59(5):3301–19.
42. Lo H-K. Method for decoupling error correction from privacy amplification. New J Phys. 2003;5(36):36–13624.
43. Ma X, Lütkenhaus N. Improved data post-processing in quantum key distribution and application to loss thresholds in device independent QKD. Quantum Inf Comput. 2012;12(3–4):203–14.